

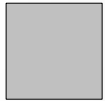
Array di Stringhe

Stringhe in C

```
char* stringa;
```

```
stringa = malloc((M+1) * sizeof(char));  
if (stringa == NULL) return -1;  
scanf("%s", stringa);
```

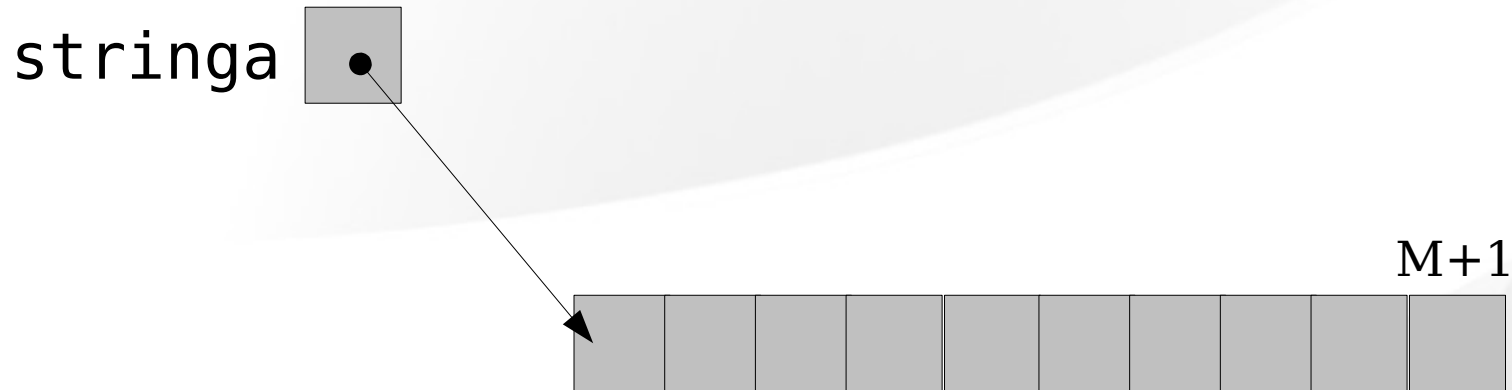
stringa



Stringhe in C

```
char* stringa;
```

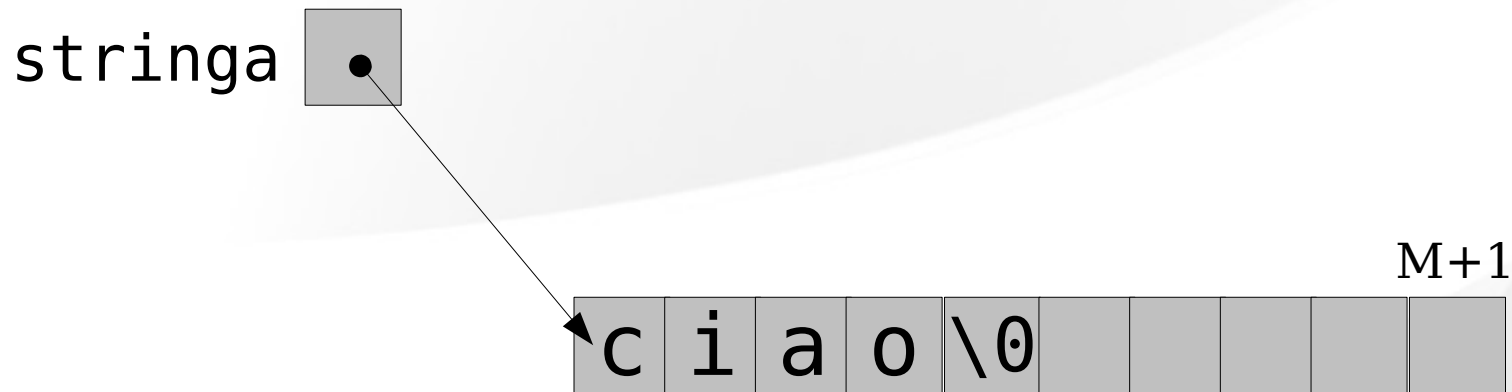
```
stringa = malloc((M+1) * sizeof(char));  
if (stringa == NULL) return -1;  
scanf("%s", stringa);
```



Stringhe in C

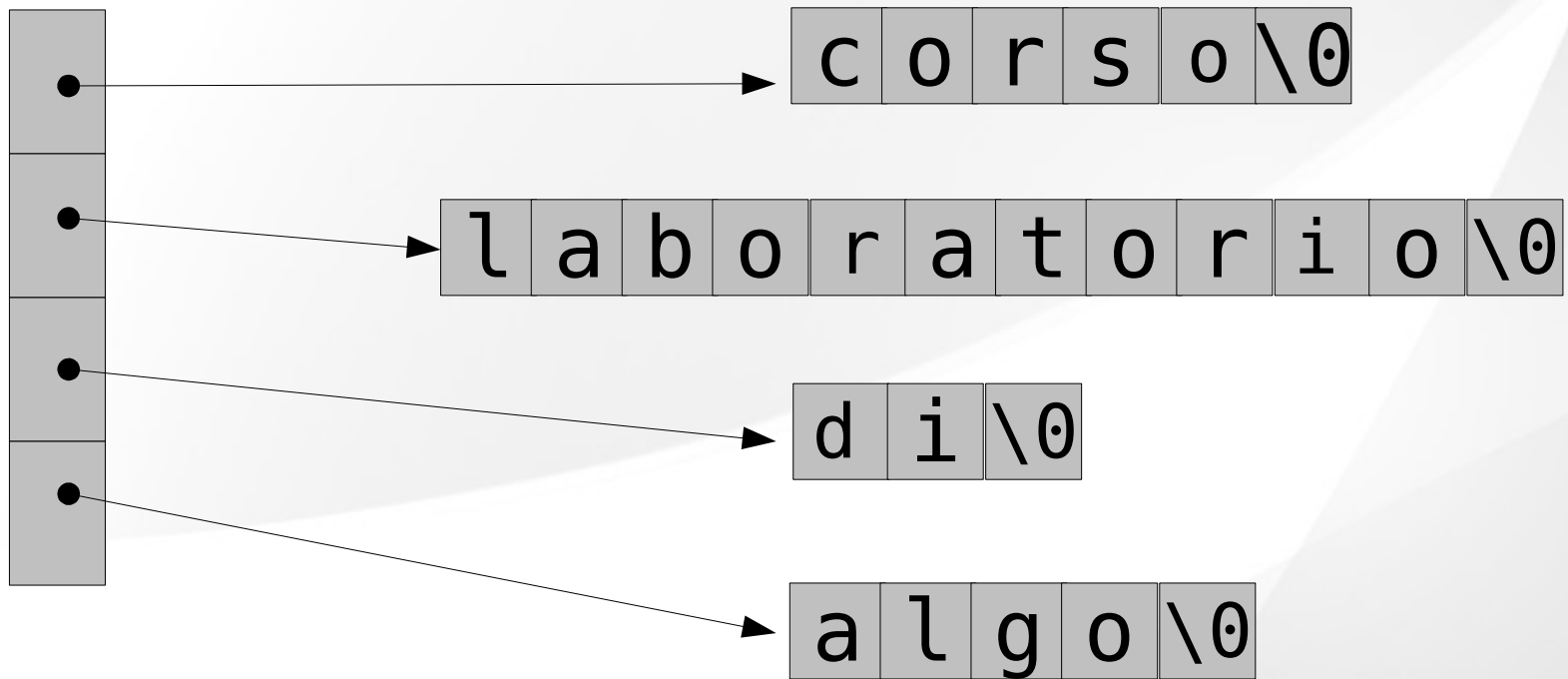
```
char* stringa;
```

```
stringa = malloc((M+1) * sizeof(char));  
if (stringa == NULL) return -1;  
scanf("%s", stringa);
```



Array di Stringhe

```
char* a[]={"corso","laboratorio","di","algo"}
```



Array di Stringhe

```
char **a;  
  
a = malloc(N*sizeof(char *));  
if (a == NULL) return -1;  
  
for(i=0; i < N; i++) {  
    a[i] = malloc((M+1) * sizeof(char));  
    if (a[i] == NULL) return -1;  
    scanf("%s", a[i]);  
}
```

Array di Stringhe

```
char **a;
```

```
a = malloc(N*sizeof(char *));  
if (a == NULL) return -1;
```

```
for(i=0; i < N; i++) {  
    a[i] = malloc((M+1) * sizeof(char));  
    if (a[i] == NULL) return -1;  
    scanf("%s", a[i]);  
}
```

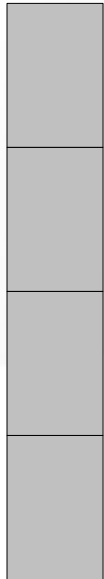
`&a[i] ?` `a+i?`

Array di Stringhe

```
char **a;
```

```
a = malloc(N*sizeof(char *));  
if (a == NULL) return -1;
```

```
for(i=0; i < N; i++) {  
    a[i] = malloc((M+1) * sizeof(char));  
    if (a[i] == NULL) return -1;  
    scanf("%s", a[i]);  
}
```

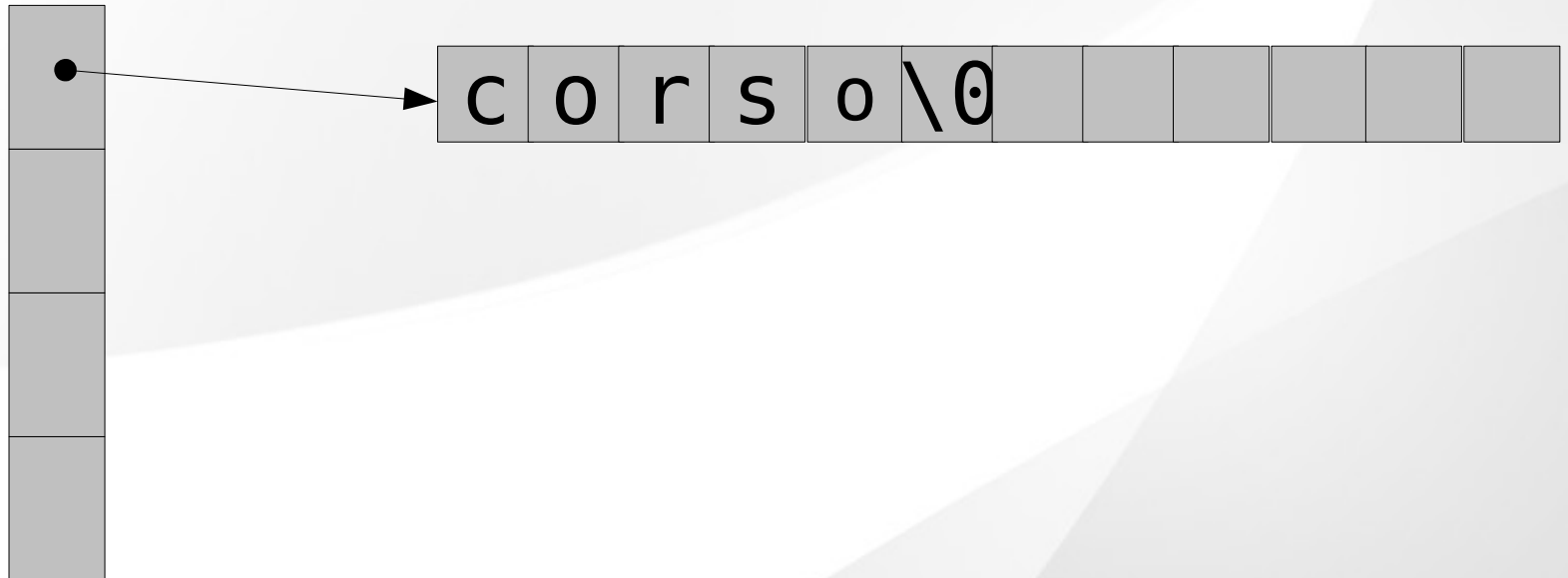


Array di Stringhe

```
char **a;
```

```
a = malloc(N*sizeof(char *));  
if (a == NULL) return -1;
```

```
for(i=0; i < N; i++) {  
    a[i] = malloc((M+1) * sizeof(char));  
    if (a[i] == NULL) return -1;  
    scanf("%s", a[i]);  
}
```



Liberare la memoria

Per rilasciare la memoria

```
for(i=0;i<n;i++){  
    free(a[i]);  
}  
free(a);
```

Prima di chiamare la free sull'array
chiamarla su ogni singola stringa